Asian Journal of Information Technology 15 (3): 627-634, 2016
ISSN: 1682-3915

An Enhanced Resource Optimization for Cloud Based Applications

[1]V. Venkatesa Kumar, [2]S. Daniel Madan Raja and [1]M. Newlin Rajkumar
[1]Department of CSE, Anna University Regional Campus Coimbatore, India
[2]Department of IT, Bannari Amman Institute of Technology, Sathyamangalam, India

Abstract: Cloud computing may be a phrase accustomed to describe a variety of competing ideas that involve an oversized range of computers connected through a period of time communication network such as the web, internet, etc. Such cloud computing allows users to utilize the computation, storage, data and services from around the world in commercialize manner. In a cloud environment, task scheduling algorithms play an important role where the aim is to schedule the tasks effectively so as to reduce the turnaround time and improve resource utilization. Many scheduling techniques have been developed by the researchers like GA (Genetic Algorithm), PSO (Particle Swarm Optimization), min-min, max-min, X-suffrage, etc. An optimized algorithm based on the fuzzy based optimization has proposed which makes a scheduling decision by evaluating the entire group of task in the job queue. The simulation results show that execution time and the response time of an application are independent of each other which are executed separately by different algorithms.

Key words: Cloud computing, scheduling algorithm, turnaround time, jfuzzy, resource

INTRODUCTION

Cloud computing is a mechanism or principle which is useful in explaining how multiple computing principles used in a network of computers and other network resources, interconnected in a real time computer network such as the internet. In internet many computing services (cloud services) are provided for the requested users. The cloud services (hardware, software and infrastructure) use by individual users, businessmen, entrepreneurs, etc., from remote locations without actually having the complete physical infrastructure in their own locations. It makes the computing resources to be pooled at particular a place and is shared based on the demands made the users from anywhere in the earth and at any time where the network connection is available. The example for such services can be storing of files at the remote location of the memory space provided by a service provider, using a software application like office 360 from remote location without installing the original software in the user owned computers. This leads to a reduction in the cost of maintaining the infrastructure and procurement of equipments for the connectivity. The cost of usage would also be minimal. So, the resource shortage would be eliminated and utilization of computing resources would be attained to the maximum extend.

The following definition of Cloud computing has been developed by the US National Institute of Standards and Technology (NIST). Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models and four deployment models. The cloud computing service models are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS).

Literature review

Scheduling: Scheduling is a process or mechanism of assigning the computing resources like printer, processor, etc for the demanding user based on the ideal or availability of the resources for computing purposes. The access to resources can be obtained through this process. Normally, it is done for load balancing purposes of a computing system in order to achieve a maximum quality of services in the form of effective utilization of the same. To do this, the need of scheduling algorithm comes into the picture for the allocation of resources because the current or the modern communication system requires

Corresponding Author: V. Venkatesa Kumar, Department of CSE, Anna University Regional Campus, Coimbatore, India

multi-tasking and multiplexing of applications for completing tasks in a rapid manner as the world necessitates. In this case, the concern comes from on the aspects of efficiency (throughput), effectiveness, latency like process time, arrival time, response time and fairness waiting time.

The real usage of cloud computing is its scalability. It means the capability of a network to add or remove the services in the form of software, hardware and infrastructure based on the demand of a company from the service provider. The demand of an IT Company or a user may vary from time to time. It needs not to be same always. So, the requirement varies. For example, a company may require a new service due to starting of a new division in the company so that some new specialists may join the company. At this juncture, the company requires a new set of software and hardware services. Hence, the demand varies. So, the scalability plays vital role in this case.

Cloud scheduler for distributed compute clouds: It explains a mechanism that shows hoe simple the usage of IaaS clouds in High Throughput Computing (HTC) environment. Every user creates his/her own virtual machine and puts this information in the VM image database and the job script that includes information about their VM. It is given as input to the condor job scheduler (Armstrong *et al*., 2010). The cloud scheduler accepts the inputs and reads the queues of the job scheduler and request the one of the available cloud resources boot the user VM. The user VM advertises itself to the job scheduler and the job scheduler in turn moves the user job to that VM. The condor job scheduler will manage the job priority and schedules the jobs accordingly. This is done based on the parameters of user fairness, resource utilization and priorities. Now the job schedules the jobs to all VM in eventual manner. As the jobs accumulate the scheduler would re-distribute the jobs by shutting down the heavily allocated VMs and moves the jobs to lightly loaded VMs.

Resource allocation for grid computing: In all computing environment, allocation or distribution of computing resources is a pivotal task because of the market requirement for resources. For this, four principles have been used by developers such as state based (current snapshot), pre-emptive (migrate to different systems), non-pre-emptive (execute in host itself) and model based (predicts system state) (Gomoluch and Schroeder, 2003). In these three algorithms like Round Robin, Proportional protocol and Continuous Double Action are used based on the parameters such as delay in message, delay in processing, creation of task, speed of the servers in use. Out of these four principle state based and non-pre-emptive are for minimizing the communication overhead but the pre-emptive system is more dynamic systemz (Gomoluch and Schroeder, 2003).

Fuzzy logic: The proposed fuzzy controller uses Fuzzy Logic (FL) (Lee, 1990; Hellendoorn *et al*., 1993; Bonissone, 1994; Yager and Filev, 1994), introduced by Zadeh (1965). The FL does not follow any strict mechanism for assignment of sets like that of binary logic. Instead, every element is assigned with some degree of membership to a set. The degree of membership is represented by a value set between 0-1. To apply this, FL system is needed to be built based on the following three stages (Lee, 1990). This is can be applied to any set of problems. The examples include scheduling of resources among the virtual machines or small cloudlets.

Falsification: In this stage, the degree of membership of the input values is assigned to Fuzzy sets. The degree of membership is given by $\mu:X\neg[0, 1]$, where X is the set of input values. So, every input value is mapped to a value between zero and one.

Inference engine: This system is a rule based system which is mapping input spaces to output spaces based on rule sets.

De-fuzzification: In this stage, a numerical output value is generated from the output set. Uncertainties are part of any system. The cloud system is not exception to it. In such situation, Fuzzy Set Theory (FST) gives an opportunity to represent the uncertainties. It means that FL is best way of treating random uncertainties when the events prediction in the real-time world is found to be difficult. To handle this situation a Fuzzy Control System (FCS) is used as it is a rule based system which helps to make decision based on the dynamic causes that come in the system so that right decision can be taken for optimal usage of the available resources in the online system. The ultimate role of the FCS is to replace a human by a rule based fuzzy rule based control system (Fig. 1).

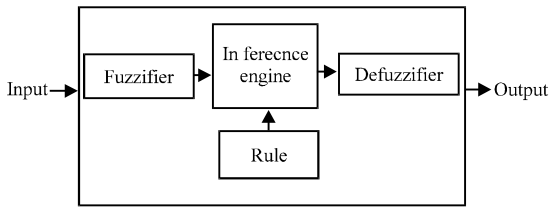

Fig. 1: Fuzzy control system

For this, at the first stage, some input variables and control variables are defined. Then, the fuzzy based rules are designed to take decisions if certain conditions are satisfied with the rules set defined by the developer. The rules are based on the IF-ELSE conditions. So, that the input variables are quantified and assigned a membership function. Based on these variables and rules output is generated according to the rules of the interface engine.

Fuzzy inference system: It is well known from the above discussion that Fuzzy Set Theory (FST) is a perfect system which could be used for concluding some decision when there is raise of uncertainties neither in random nor static form (Prasath *et al*., 2013). Based on this, the Fuzzy Interface System (FIS) is being used to evaluate the cloud computing user's satisfaction. It (FSI) is basically a rule based system that uses FST and fuzzy logic. The FSI takes into account the input spaces given by the user and generates into output spaces. The input variables are taken into consideration from the stored values in the system. These values are evaluated based on the set of rules framed according to fuzzy logic. So, FSI would generate responses from the system that may help in opting right kind of decision. This might result in generation of expected result. In this entire process a member function would act as curve that defines how the fuzzy variables are mapped in a degree of membership from 0-1. This system is also considered to be deductive form as the decision taken based on IF-ELSE format like human do. The Fuzzy Inference system is a popular way for wide range of science and engineering. In step two, for making rules the verbal options of experts regarding the effects of different factors such as security, efficiency and performance, adaptability and cost are gathered and processed for generating a rule base and using them as inputs of our Fuzzy Inference system.

Existing system: Change is permanent. The information technology regime is not an exception to this principle. In computational process the demand of people to complete their task on-demand while on fly. To do this, the role cloud computing comes into the picture to fulfill the needs of the people. So, it the cloud computing that is considered to be the next generation computational mechanism for the people because in the rapidly developing world that experience globalization requires a consolidated tool or mechanism that should be used for development and deployment of increased number of applications like huge space for data storage, software for dynamic application according to the needs of the business. The solution to this is cloud computing. The best example in this aspect is Google's MapReduce (Dean and Ghemawat, 2004). Hadoop MapReduce running on top of Hadoop Distributed File System (HDFS) is inspired by Google MapReduce. Hadoop divides works with a Map function and a Reduce function into map tasks and reduce tasks. Job scheduling in Hadoop is performed by a master node which receives pulses sent by slaves every few seconds. Each slave has a fixed number of map slots and reduces slots to execute map or reduce tasks. These tasks are parallel processed on the nodes of the cluster by the policy which strives to keep the work as close to the data as possible.

In this, the role of jobs (works) scheduling comes into forefront as it is so vital because it related with the efficiency of the cloud system in a distributed computing network. Hence, the scheduling needs to spread the workload (jobs) in a balanced manner to the required system (node) based on the pulses it receives from the respective nodes in the network. This would result in maximizing the resource utilization while minimizing the execution time (Zomaya and Tan, 2001). Task scheduling, one of the most combinational mechanisms that help for optimization, plays important role in improving the flexibility as well as reliability of the cloud system. In this the mechanism needs to find the right sequence of the demands made by the nodes in accordance with the adaptable time for task scheduling based on the set out constraints (logic). Based on the approaches adopted in the system, it has been divided into two:

- Dynamic load balancing
- Static load balancing

The dynamic load balancing mechanism is considered to be good when compared to static load balancing system as the former does all the tasks in real-time than the later. But, the former has the problem of higher overhead as it required updating the status continuously on fly. Current Hadoop scheduler includes FIFO, FAIR scheduler (Zaharia *et al*., 2009) and Capacity scheduler. Genetic Algorithms (GA) (Tayal, 2011) with different components are developed on the based technique for task level scheduling in Hadoop MapReduce. To achieve a better balanced load across all the nodes in the cloud environment, they revise the scheduler by predicting the execution time of tasks assigned to certain processors and making an optimal decision over the entire group of tasks.

Disadvantages of existing system:

- The high computing power required for resource mapping on preferential basis
- System information needs to be updated continuously
- Dynamic load balancing has higher overhead compared to static load balance.
- Genetic algorithm faces problem in resolving few optimization problems (they are called variant problems)

The basically genetic algorithm performs well with good chromosome block but it could not resolve those few optimization problem due to formation of poor chromosome by the fitness function (Jose *et al*., 2009; Bitam, 2012).

Problem defination: The right way of choosing an algorithm for application execution is a major task in today's world requirement of utilizing available computational resources. In this respect the fuzzy Logic system helps to identify right algorithm based on rules based system though, we find algorithms like Genetic algorithm, etc., to optimize scheduling of task. Because each algorithm has its own drawbacks. The jfuzzy logic system takes execution time and response time into consideration in deciding the decision.

MATERIALS AND METHODS

Proposed system: The proposed system is designed based on the components of fuzzifier, De-fuzzifier and FSI according to the rules set by us to arrive at a solution which is optimal according to the availability of the computational resources in the system.

Advantages of proposed system: In realtime, taking right decision is not a simple task as people consider. Because uncertainities are part of the current world. To avoid such problem, we are going to use jfuzzy logic. This would handle the situation based on the current availability of input parameters. It has the following features which have advantage over the previous existing system:

- Standardization reduces programming work
- It follows the object-oriented approach
- Its platform-independent language

System architecture: The system architecture is shown below shows the entire process of system design (Fig. 2). The choosing of algorithm for scheduling has been decided based on the input and output requirements of the jFuzzy Logic system.

The system flow diagram clearly explains the entire system of working principle of choosing of right algorithm based on the given input variables like response time and execution time (Fig. 3). As explained earlier the IF-ELSE framework is being used to decide what kind of algorithm has to be opted from the given four algorithm based on the rule sets that entirely depends on the MAX_Execution Time and MIN_Execution Time, MAX_Response Time and MIN_Response Time. Based on these variables along with the jFuzzy rules set the right decision is taken to choose the algorithm for scheduling.

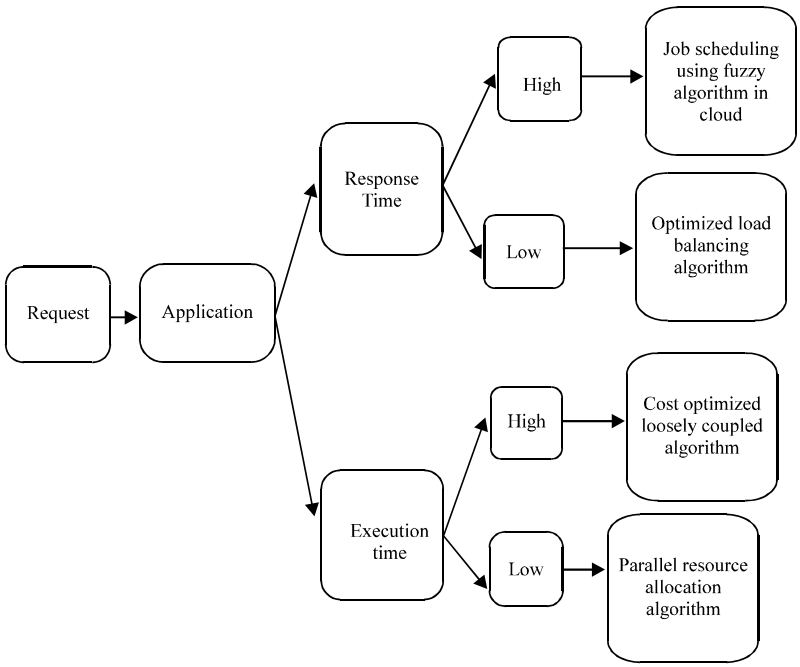

Fig. 2: System design

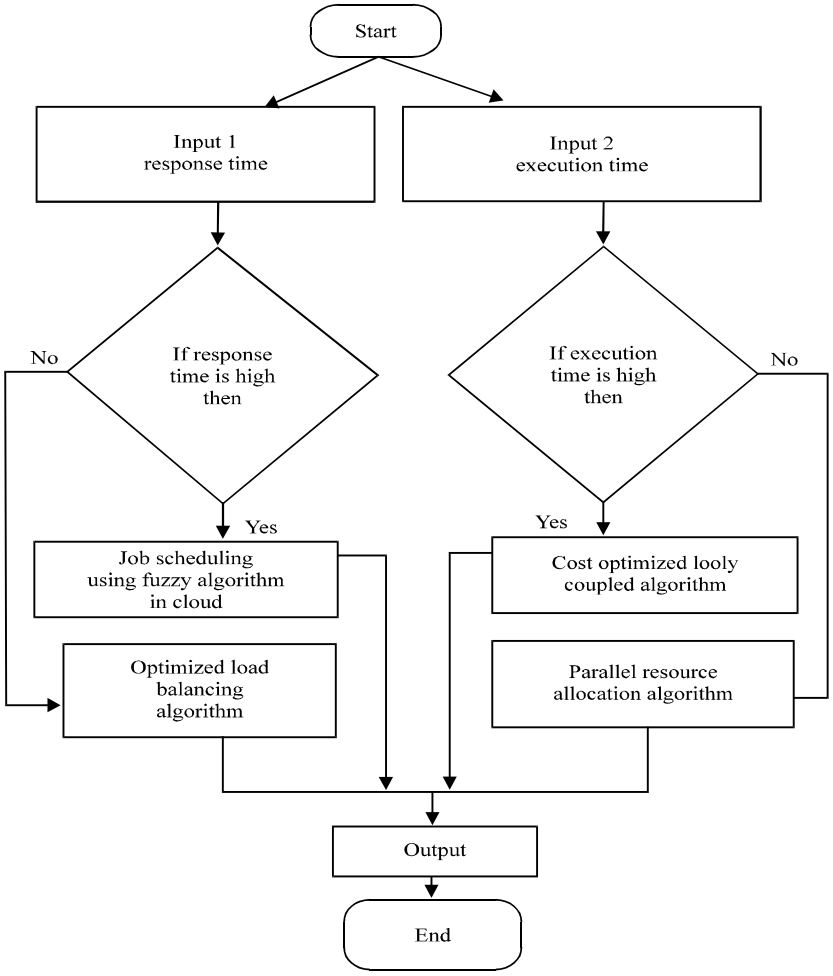

Fig. 3: System flow diagram with fuzzy architecture

RESULTS AND DISCUSSION

The empirical results show that what kind of outputs are being generated when the input variables are set out by the member function. As a result when the response time and execution time is low, the optimized load balancing algorithm and parallel resource allocation algorithm are being chosen respectively. The optimized results for those algorithms are shown in Fig. 4 and 5. If those times are high then the other two algorithms job scheduling using fuzzy and cost optimized loosely coupled algorithm are chosen for high response and execution in a respective ways. The outputs for these algorithms are shown in Fig. 6 and 7, respectively.

Table 1 and 2 shows the sample values of response time, execution time and optimized algorithm when response time is low. Table 3 and 4 shows the sample values of response time, execution time and optimized

Table 1: Sample value of response time when it's become low

Response time range	Response time value	Algorithm range (%)
0.0	LOW	7.50
1.0	LOW	7.50
2.0	LOW	7.50
3.0	LOW	7.50
4.0	LOW	7.50
5.0	LOW	7.51

Table 2: Sample value of execution time when response time is low

Execution time range	Execution time value	Algorithm range (%)
0.0	NIL	7.50
1.0	NIL	7.50
2.0	NIL	7.50
3.0	NIL	7.50
4.0	NIL	7.50
5.0	NIL	7.51

Table 3: Sample value of response time when it's become high

Response time range	Response time value	Algorithm range (%)
15.1	HIGH	2.50
16.0	HIGH	2.50
17.0	HIGH	2.50
18.0	HIGH	2.50
19.0	HIGH	2.50
19.9	HIGH	2.50

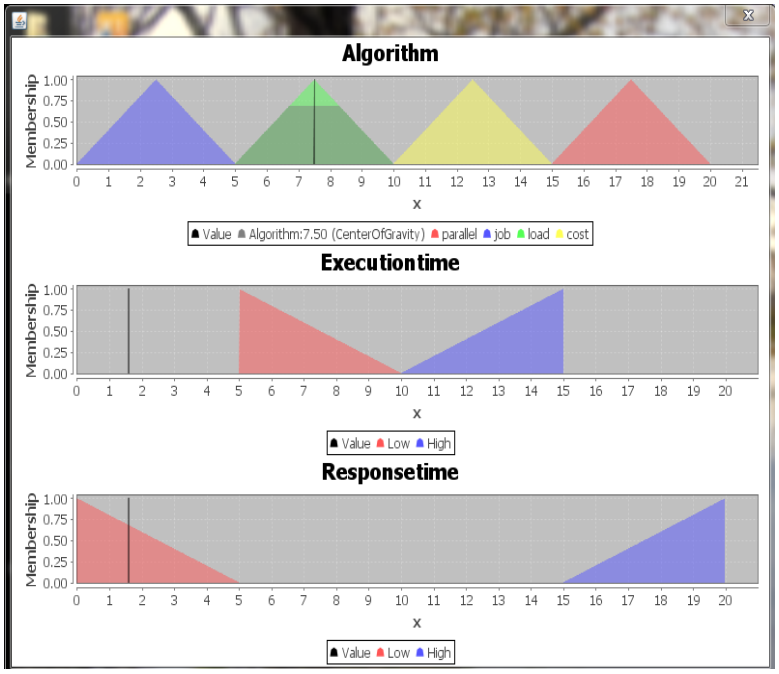

Fig. 4: Sample output when response time is low

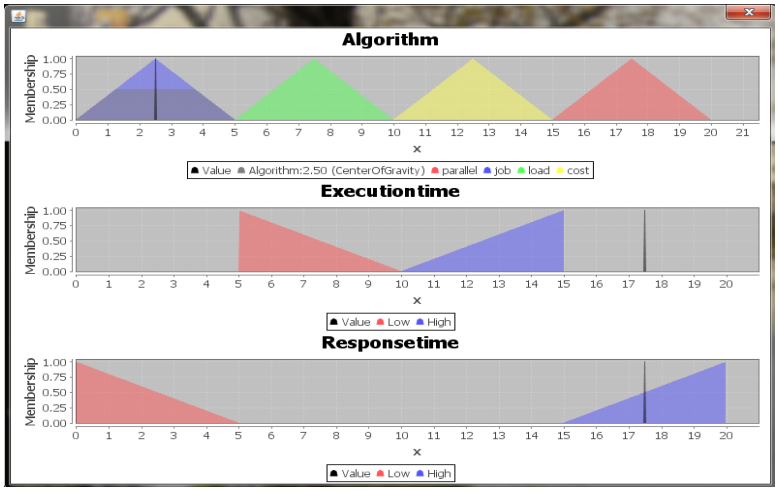

Fig. 5: Sample output when response time is high

Table 4: Sample value of execution time when response time is high

Execution time range	Execution time value	Algorithm range (%)
15.1	NIL	2.50
16.0	NIL	2.50
17.0	NIL	2.50
18.0	NIL	2.50
19.0	NIL	2.50
19.9	NIL	2.50

Table 5: Sample value of response time when execution time is low

Response time range	Response time value	Algorithm range (%)
5.1	NIL	17.50
6.0	NIL	17.50
7.0	NIL	17.50
8.0	NIL	17.50
9.0	NIL	17.50
10.0	NIL	17.50

Algorithm when response time is high. Table 5 and 6 shows the sample values of response time, execution time and optimized algorithm when execution time is low.

Similarly Table 7 and 8 shows the sample values of Response time, Execution time and optimized Algorithm when Execution Time is high (Table 9 and 10).

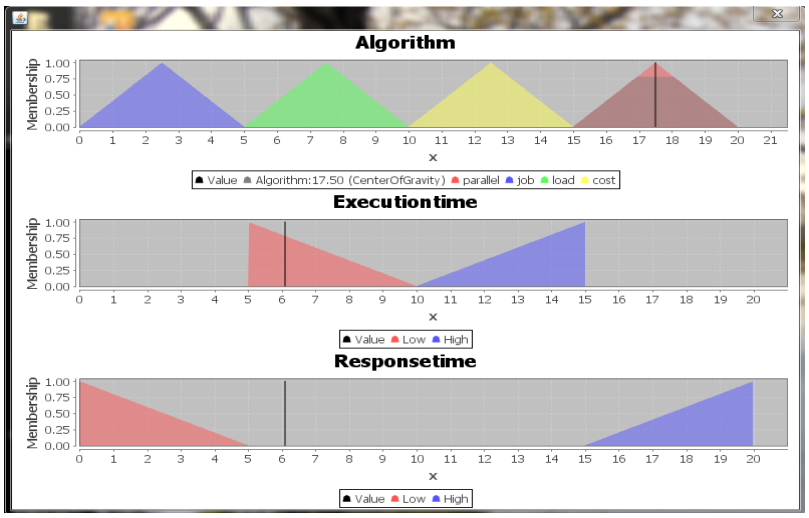

Fig. 6: Sample output when execution time is low

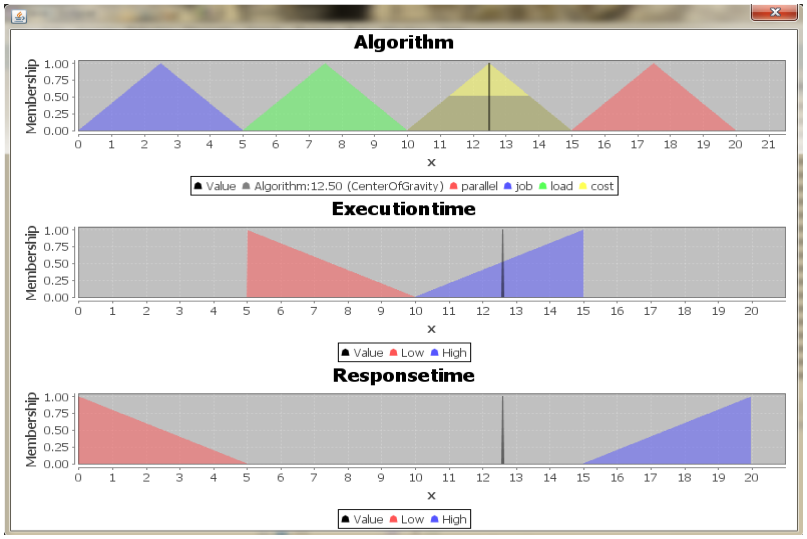

Fig. 7: Sample output when execution time is high

Table 6: Sample value of execution time when it's become low

Execution time range	Execution time value	Algorithm range (%)
5.1	Low	17.50
6.0	Low	17.50
7.0	Low	17.50
8.0	Low	17.50
9.0	Low	17.50
10.0	Low	17.50

Table 7: Sample value of response time when execution time is high

Response time range	Response time value	Algorithm range (%)
10.1	Nil	12.50
11.0	Nil	12.50
12.0	Nil	12.50
13.0	Nil	12.50
14.0	Nil	12.50
15.0	Nil	12.50

Table 8: Sample value of execution time when it's become high

Execution time range	Execution time value	Algorithm range (%)
10.1	High	12.50
11.0	High	12.50
12.0	High	12.50
13.0	High	12.50
14.0	High	12.50
15.0	High	12.50

Table 9: Selection of algorithm based on response time

Response time range	Response time value	Optimized algorithm
1.0	Low	Optimized load balancing algorithm
6.0	Nil	Parallel resource allocation algorithm
11.0	Nil	Cost optimized loosely coupled algorithm
16.0	High	Job scheduling using fuzzy algorithm in cloud

Table 10: Selection of algorithm based on execution time

Execution time range	Execution time value	Optimized algorithm
1.0	Nil	Optimized load balancing algorithm
6.0	Low	Parallel resource allocation algorithm
11.0	High	Cost optimized loosely coupled algorithm
16.0	Nil	Job scheduling using fuzzy algorithm in cloud

CONCLUSION

A system has been designed in order to decide on the algorithm selection which would help for any Application to opt for the available computational resources of any system. For this the jfuzzy logic system has been used with rule set in order to take right decision for any application even during uncertainties.

The system which has been designed needed to generate its decision based on the rules. At the same time, it is considering only two variables based on predefined cases. If these parameters increased by considering the real-time situation. It would be usefull in decision making for effective utilization of resources in the resource contrained world.

REFERENCES

Armstrong, P., A. Agarwal, A. Bishop, A. Charbonneau and R. Desmarais *et al*., 2010. Cloud scheduler: A resource manager for distributed compute clouds. http://arxiv.org/pdf/1007.0050.pdf.

Bitam, S., 2012. Bees life algorithm for job scheduling in cloud computing. Proceedings of the 3rd International Conference on Communications and Information Technology, June 26-28, 2012, Hammamet, Tunisia, pp: 186-191.

Bonissone, P.P., 1994. Fuzzy Logic Controllers: An Industrial Reality. In: Computational Intelligence: Imitating Life, Marks II, R.J., J.M. Zurada and C.J. Robinson (Eds.). IEEE Press, USA., ISBN-13: 978-0780311046, pp: 316-327.

Dean, J. and S. Ghemawat, 2004. MapReduce: Simplified data processing on large clusters. Proceedings of the 6th Symposium on Operating Systems Design and Implementation, December 6-8, 2004, San Francisco, CA., USA., pp: 137-150.

Gomoluch, J. and M. Schroeder, 2003. Market-based resource allocation for grid computing: A model and simulation. Proceedings of the 1st International Workshop on Middleware for Grid Computing, (MGC'3), London, UK., pp: 211-218.

Lee, C.C., 1990. Fuzzy logic in control systems: Fuzzy logic controller. II. IEEE Trans. Syst. Man Cybernet., 20: 419-435.

Prasath, V., N. Bharathan, N.P. Neetha, N. Lakshmi and M. Nathiya, 2013. Fuzzy logic in cloud computing. Int. J. Eng. Res. Technol., 2: 1-5.

Tayal, S., 2011. Tasks scheduling optimization for the cloud computing system. Int. J. Innov. Eng. Technol., 5: 111-115.

Yager, R.R. and D.P. Filev, 1994. Essentials of Fuzzy Modeling and Control. 1st Edn., Willy, New York, ISBN-13: 978-0471017615, Pages: 408.

Zadeh, L.A., 1965. Fuzzy sets. Inform. Control, 8: 338-353.

Zaharia, M., D. Borthakur, J.S. Sarma, K. Elmeleegy, S. Shenker and I. Stoica, 2009. Job scheduling for multi-user mapreduce clusters. Electrical Engineering and Computer Sciences University of California at Berkeley, Technical Report No. UCB/EECS-2009-55, April 30, 2009.

Zomaya, A.Y. and Y.H. Tan, 2001. The observations on using genetic algorithms for dynamic load balancing. IEEE Trans. Parallel Distrib. Syst., 12: 899-911.